

MÔN: PHÂN TÍCH THIẾT KẾ THUẬT TOÁN

PHÂN TÍCH BÀI TOÁN NUMBER PARTITIONING

23rd January 2021

GIẢNG VIÊN: PHẠM NGUYỄN TRƯỜNG AN

Võ Huy Khôi - 18520949
Hứa Văn Sơn - 18521344
Nguyễn Thịnh Quyền - 18521322

Contents

1	Giới thiệu bài toán	2
1.1	Mô tả hình thức	2
1.2	Ứng dụng bài toán	2
1.3	Một ứng dụng thực tế	2
2	Phương pháp thiết kế thuật toán	2
2.1	Mã giả	3
2.2	Tính độ phức tạp bằng lý thuyết	3
2.3	Mã nguồn cài đặt	4
2.4	Phân tích input, output kiểm tra độ đúng đắn	4
2.5	Phân tích độ phức tạp thuật toán bằng thực nghiệm	4
2.6	Tham khảo	5

1 Giới thiệu bài toán

Partition hay number partitioning là bài toán xem xét 1 tập hợp số nguyên dương có thể chia thành 2 tập hợp con có tổng bằng nhau hay không.

1.1 Mô tả hình thức

- Input: list các số nguyên S , n số phần tử trong S
- Output: True nếu S có 2 tập hợp S_1, S_2 với $\text{sum}(S_1) = \text{sum}(S_2)$. False nếu S không có.
Điều kiện: $n * (\text{sum}/2) < 10^9$

1.2 Ứng dụng bài toán

Bài toán number partitioning có rất nhiều ứng dụng như:

- Ứng dụng trong phân chia công việc.
- Thao túng các cuộc bầu cử.

1.3 Một ứng dụng thực tế

Khi phân chia lượng công việc cho nhóm với một số lượng công việc nhất định, áp dụng number partitioning để tìm ra lượng công việc bằng nhau để chia cho 2 người, nếu không có lượng công việc bằng nhau có thể nâng cấp bài toán lên thành chia công việc sao cho tỉ lệ chênh lệch công việc là thấp nhất.

2 Phương pháp thiết kế thuật toán

Dynamic programming

Dynamic programming được phát minh bởi nhà toán học Richard Bellman vào năm 1953 là phương pháp giúp tối ưu thời gian chạy cho những bài toán có dạng bài toán con gối nhau (overlapping subproblem) và cấu trúc con tối ưu (optimal substructure).

Giải quyết bài toán bằng cách chia bài toán thành các bài toán con nhỏ hơn và lưu lại kết quả cho những lần tính toán tiếp theo. Từ đó giải quyết cho toàn bộ bài toán.

Ý tưởng: Áp dụng phương pháp của bài toán subset sum cho number partitioning. Nếu trong list ban đầu có subset có tổng bằng $\text{sum}/2$ với sum là tổng tất cả phần

tử trong list. Tạo một mảng 2 chiều với hàng là subset trong list theo thứ tự và cột là tổng từ 1 đến $\text{sum}/2$.

Ưu điểm: tiết kiệm thời gian tính toán.

Nhược điểm: tốn nhiều bộ nhớ với việc các phần tử trong list có giá trị càng lớn hoặc càng nhiều phần tử thì mảng 2 chiều càng lớn.

2.1 Mã giả

Parttion(arr, n):

1. $\text{sum} = 0$
2. for $i = 0, \dots, n$: $\text{sum} += 1$
3. if $\text{sum} \div 2 \neq 0$: return False
4. Initialize $M[i][j] = \text{True}$ for $i = 0, \dots, n+1$, for $j = 0, \dots, \text{sum}/2 + 1$
5. $M[i][0] = \text{True}$ for $i = 1, \dots, \text{sum}/2 + 1$
6. for $i = 1, \dots, \text{sum}/2 + 1$
 7. for $j = 1, \dots, n+1$
 8. $M[i][j] = M[i][j-1]$
 9. if $i \geq \text{arr}[j-1]$: $M[i][j] = M[i][j] \vee M[i-\text{arr}[j]][j-1]$
10. return $M[\text{sum}/2][n]$

2.2 Tính độ phức tạp bằng lý thuyết

- Tại dòng 1, 3: phép gán và phép so sánh \Rightarrow độ phức tạp là 3
- Tại dòng 2: vòng lặp chạy từ $0 \rightarrow n$ và thực hiện phép cộng mỗi vòng lặp \Rightarrow độ phức tạp thời gian n
- Tại dòng 4: Khởi tạo gán M trong 2 vòng lặp n và $\text{sum}/2 \Rightarrow$ độ phức tạp $n * \text{sum}/2$
- Tại dòng 5: vòng lặp chạy từ $1 \rightarrow \text{sum}/2 + 1$ và thực hiện phép gán \Rightarrow độ phức tạp $\text{sum}/2$
- Tại dòng 6,7,8,9: phép gán dòng 8 có độ phức tạp 1, phép so sánh dòng 9 có độ phức tạp 2 vì trong điều kiện có phép gán và phép logic. Dòng 8, 9 lặp lại $\text{sum}/2 * n$ lần \Rightarrow độ phức tạp $n * \text{sum}/2$

Áp dụng quy tắc cộng: độ phức tạp của thuật toán là $n \cdot \text{sum}/2 + n \cdot \text{sum}/2 + n + \text{sum}/2 + 3$.

Áp dụng qui tắc bỏ hằng số và lấy Max: độ phức tạp thời gian là $O(n \cdot \text{sum})$.

Độ phức tạp không gian: $O((n+1) \cdot (\text{sum}/2 + 1)) = O(n \cdot \text{sum})$

2.3 Mã nguồn cài đặt

2.4 Phân tích input, output kiểm tra độ đúng đắn

- Mảng rỗng hoặc chỉ có 1 phần tử: return False
- Mảng có tổng lẻ: return False
- Mảng có n phần tử bằng nhau: nếu n lẻ return False, n chẵn return True

2.5 Phân tích độ phức tạp thuật toán bằng thực nghiệm

size	sum//2	n	operations	297176.98605673*sqrt(n)	MSE	5.22949542e+08*log(n)	MSE	17.00518159*n	MSE	0.86625649*nlog(n)	MSE	4.70586713e-08*n^2	MSE
13	13196	171548	2929817	123085814.4	123050940.1	9093178828	9093178356	2917204.891	271557.13	2583973.799	1380908.057	1384.876287	2929816.673
14	14976	209664	3579591	136074612.8	136027522.2	9244555390	9244554697	3565374.393	318711.728	3210676.952	1582727.153	2068.651797	3579590.402
15	24947	374205	6386781	181789990.5	181677763.3	9681610296	9681608190	6363423.977	545716.806	6001280.11	2185316.586	6589.596661	6386777.601
16	28295	452720	7724906	199953962.2	199804686.7	9825311709	9825308673	7698585.809	637141.463	7368222.513	2320230.53	9644.928724	7724899.979
18	43487	782766	13350924	262924556.8	262585367.9	10238422002	10238413297	13311077.97	1030715.7	13275524.67	1416903.836	28833.91194	13350892.86
26	102572	2666872	45439987	485306669.3	483174679.5	11163255958	11163163476	45350642.64	2848092.51	49315083.03	19162071.8	334690.9768	45438754.39
33	126079	4160607	70857143	606168719.6	602013107.8	11498804146	11498585829	70751877.56	3860898.33	79249418.95	35492755.46	814616.217	70852460.19
36	161439	5811804	98962918	716424870	709556858.4	11750966246	11750549522	98830782.39	5112298.13	113128297.9	54811975.3	1589503.834	98950152.18
42	169753	7129626	121374338	793502954.5	784165294.3	11905152872	11904534144	121240584.8	5696536.02	140601009.7	70971219.52	2392065.998	121350764.1
46	243107	11182922	190353812	993785879.6	975384950	12244755728	12243276043	190167619.3	8417262.4	226825633.3	123350294.2	5885051.29	190262818
50	296047	14802350	251937116	1143353454	1115251008	12456304000	12453755941	251716649.7	10537489.2	305426406.2	172664354.1	10311005.02	251726028.8
60	383752	23025120	391812131	1425988834	1371104449	12789622856	12783619834	391546346.7	14429288.3	487805120.4	290580607.6	24948444.05	391017034.3
86	623764	53643704	912568643	2176579591	1976035725	13427725982	13396680317	912220927.7	25189428.1	1193185431	76706670.1	135418243.3	902465194.7
94	714746	67186124	1142880941	2435874827	2151181858	13597556171	13549441205	1142512239	29028075.3	1513307426	991928686.4	212421677.9	1122966552
124	918252	113863248	1996596215	3171078243	25110402079	13995555784	13860922654	1936265209	35804212.8	2639735616	1793822517	610108108.5	1837980684
163	1371220	223508860	3801025445	4442859153	2300261512	14504401306	13997494876	3800808751	40586571.4	5370086300	37934219623	2350872889	2968836302
196	1888969	370237924	6295938008	5718151069	2634688550	14885171670	13488124416	6295963129	17785463.7	9128963365	6610532256	6450620091	1404159593
150	71810	10771500	183190629	97533781.7	957975562.9	12216475645	12215102062	183171313.5	2660161.65	217976077.8	118130283.7	5459992.326	183109243.5
181	181404	32834124	558365513	1702855870	1608709442	13057366375	13045422362	558350241	4129711.62	710178856.8	438841614.4	50732998.18	556055940.5
218	333182	72636676	1235110489	2532702541	2211127368	13656375075	13600407430	1235148850	9734548.97	1643085545	1083619947	248265121.1	1209901793
					1164737751		11951707166		10931194.1		818746048.1		570352264.6

trong đó: sum: tổng các phần tử trong list

size: số phần tử trong list

n: sum*size

MSE:

- $\sqrt{\text{sum} * \text{size}} = 1164737751$
- $\log(\text{sum} * \text{size}) = 11951707166$
- $\text{sum} * \text{size} = 10931194.11$
- $\text{sum} * \text{size} * \log(\text{sum} * \text{size}) = 818746048.1$
- $(\text{sum} * \text{size})^2 = 570352264.6$

Có thể thấy MSE của sum*size là bé nhất, như vậy độ phức tạp của thuật toán theo thực nghiệm là $O(\text{sum} * \text{size})$, bằng với độ phức tạp phân tích lý thuyết.

Thuật toán tối ưu hóa không gian

Thay vì tạo một mảng 2 chiều có kích thước bằng $(size+1) * (sum/2 + 1)$ Tạo một mảng với số phần tử bằng $sum/2 + 1$ Phần tử thứ j sẽ là True có tập hợp con có tổng bằng j , ngược lại là False. Do đó độ phức tạp không gian là $O(sum)$.

part[j] = True if there is a subset with sum equal to j, otherwise False.

2.6 Tham khảo

1. <https://www.geeksforgeeks.org/partition-problem-dp-18>
2. https://en.wikipedia.org/wiki/Partition_problem

Trace back tìm subset

Bài toán nâng cấp từ number partition: Áp dụng bài toán trên để tìm ra 2 subset có tổng bằng nhau. Thực hiện traceback từ ma trận đã tạo của number partition.

- Input: ma trận partition part, tổng các phần tử trong list sum, list S
- Output: 2 subset có tổng bằng $sum/2$.